基于改进 Qsplat 算法的三维动态积云模拟研究 *

彭晏飞，米　扬，訾玲玲，赵全颖

(辽宁工程技术大学 电子与信息工程学院，辽宁 葫芦岛 125105)

摘　要：尽管人们对云模拟进行了大量研究，但是现有的云模拟算法很难在保证实时性的情况下得到真实感较强的云图形。针对这个问题，提出一种结合 Qsplat 算法与 IFS 算法的粒子系统的积云模拟方法，并考虑了积云的结构特点。首先，基于 Qsplat 算法的细节层次树型结构建立树型结构粒子系统，加速粒子系统搜索速度；然后，对距离视点较近的粒子使用 IFS 迭代函数系统算法增加层次包围球树的层次以丰富云体局部细节来建立云的外形；最后，使用球形图元映射二维纹理进行云的绘制。最终实现了三维动态积云的快速实时模拟。实验结果表明，该方法能够快速地生成实时三维动态积云图形，图形真实感强。

关键词：动态云模拟；粒子系统；Qsplat 算法；层次包围球树；迭代函数系统

中图分类号：TP391.9　　doi: 10.3969/j.issn.1001-3695.2017.11.0770

Research on three-dimensional dynamic cumulus simulation based on improved Qsplat

Peng Yanfei, Mi Yang, Zi Lingling, Zhao Quanying

(*School of Electronic & Information Engineering Liaoning Technical University*, *Huludao Liaoning* 125105, *China*)

Abstract: Although people have done a lot of research on cloud simulation, the existing cloud simulation algorithm can hardly obtain the cloud image with strong realism while ensuring the real-time performance. According to this problem, this paper proposes a cumulus-based simulation method for particle systems that combine Qsplat with IFS. First, build a tree-structured particle system based on the detail-level tree structure of Qsplat for speeding up particle system search speed. Then, for these particles which closer to the viewpoint , used the iterative function system algorithm to increase the level of enclosing the ball tree to enrich the local detail of the cloud to establish the shape of the cloud. Finally, Finally, rendered the cloud by using two-dimensional texture mapping of spherical entities, which realizes the fast real-time simulation of dynamic cumulus. The experimental results show that the method can generate real-time three-dimensional dynamic cumulus images quickly.

Key Words: dynamic cloud simulation; particle system ; Qsplat; levels of detail; iterated function system

0　引言

不规则模糊物体的模拟长时间以来是计算机图形学研究的主要热点和难点。云是一种广泛存在的自然不规则模糊物体，无论是影视或动漫制作还是游戏创作抑或是飞行场景仿真等等领域都无法绕开这个问题，高效而又逼真的云模拟技术对图形学发展和现代影视游戏工业以及军工及航空业都有着重要的意义。自然界云的种类众多，但很多都不常见，在大多数情况下，完全模拟所有种类没有必要。积云是自然界最常见也是特征最明显的云，完成积云的模拟基本上可以满足绝大部分需求。

积云最明显的特征有以下几种：结构紧密，外形不规则，呈“花椰菜”状；云块孤立，具有明显的自相似性；大部分云块边界明显，存在一些云块破裂，边界较为模糊[1]。这些特点就是积云模拟的难点所在。在所有现存的云模拟方法中，基本分为两类。一种是基于物理模型的模拟方法，典型的有，Harris 等通过在 N-S 方程中引入热力学方程，使用三维平铺纹理技术实现对云的真实感模拟[2]；唐勇、赵丽文等人则通过在 N-S 方程中添加粘性项与涡旋，同时考虑温度、密度等物理特性，结合单项散射光模型对三维动态云进行模拟[3]；何晓曦，陈雷霆等人通过简化 N-S 方程建立云的运动模型，同时利用 Impostor 技术配合 Harris 的云光照数学模型加速渲染大规模的云层图形[4]。这些物理模型模拟方法往往能取得较好的真实感云图形，但是因为运用了大量的物理学原理，导致运算过程非常复杂，运算量空前巨大，对计算机硬件的要求过高，并不适合于实时模拟，并且由于当前物理模型问题，这些模拟方法对云体的运动控制都存在问题。另一种模拟方法就是基于个体生长模型的云彩模

基金项目：辽宁省教育厅科学研究一般项目（L2015225）；辽宁省博士科研启动基金（201601365）

作者简介：彭晏飞（1975-），男，黑龙江五常人，副教授，博士研究生，主要研究方向为图像分析与理解、计算机视觉；米扬，男，湖南怀化人，硕士研究生，主要研究方向为计算机图形学、计算机视觉；訾玲玲（1981-），女，讲师，博士，辽宁阜新人，主要研究方向为视觉与信息获取.赵全颖，男，辽宁大连人，硕士研究生，主要研究方向为数据挖掘、数据库.

拟方法，包括很多近似模拟方法。例如贺怀清、刘浩瀚等人对这种方法进一步改进，通过利用改进的 Gardner 椭球模型结合叠加纹理的方法实现了实时的云模拟[5]。基于粒子系统的模拟方法一直认为是不规则物体模拟领域中最成功的方法。潘秋羽等人利用粒子系统结合纹理映射的方法，并利用 Impostor 等技术加快绘制，从而实现了三维动态云的实时模拟[6]，Elhaddad 利用分子动力学模拟模型[7]实现了三维云的模拟。这些方法一般能取得较为逼真的效果和较好的控制能力，但是普遍存在的问题是当模拟物体精度要求较高时往往需要大量的粒子，这对模拟的效率是严峻的考验。还有的就是传统的基于动态网格的模拟方法，例如范晓磊，张立民等人利用 N-S 方程构建动态网格，结合多向散射模型，同时采用 GPU 加速的云模拟方法[8]以及 Yusov 等提出的网格模型[9]。这些方法的模拟速度较快，但是建模过程非常复杂，当模拟云体精度较高时，建模的难度更是极具上升，同时还存在的问题是，模拟的云体是中空的，实现动态效果的难度很大，容易导致模拟失真。除了两大类研究方向，还有一些其他方法，例如考虑用户的需求生成云模拟模型方法，Dobashi 结合用户需求和真实云层图片生成云的模拟图形[10-11]，Goswami 等人引入用户设置的拉格朗日三维云模拟框架[12]；Rui 等结合气象观测数据与简单物理模型的三维云模拟方法[13]。并且近些年来，GPU 的迅速发展为动态云模拟提供了新的研究思路。GPU 高速并行的特点，对物理模型计算和碰撞检测都有非常明显的加速效果[14-17]。例如 Barbosa 等人提出的基于坐标的流体模拟技术结合 GPU 加速计算技术方法[18]。

针对积云的特点要求以及现有算法存在的不足，本文提出了一种三维动态积云的实时模拟方法。通过建立基于改进动态 Qsplat 算法层次包围球树结构粒子系统，结合迭代函数系统算法对层次包围球树树的局部粒子进行分形处理，得到细节层次更高的粒子数据，最终利用球形图元映射二维纹理绘制出积云图形。经过实验，算法在真实感和绘制速度方面取得了较好的平衡。

1 Qsplat 方法及 IFS 算法的原理

2000 年，加利福尼亚大学图形学实验室提出了一种改进的大规模点云数据加速处理算法——Qsplat 算法。这是一种基于点的绘制算法，相比于 splat 等传统基于点的算法，该算法具有更快的绘制速度。该算法的原理如下。

Qsplat 的预处理，通过长轴二分法等方法对点云数据进行划分处理，将点云树构建为层次包围球的树状结构。该树结构包含了不同细节层次的点模型，树的根节点代表整个模型，树中的各层次节点代表不同精度下的点模型采样，每个叶子节点代表一个或多个点云模型中的点数据。树中各节点最为一个基础绘制图元，其属性包括位置、半径、法线以及颜色等。树中节点的属性计算要自底向上进行。在包围球树结构建立后，进行遍历绘制。遍历标准：从根节点开始，判断当前节点包围球的屏幕投影是否大于阈值。若大于阈值，则继续向下遍历；若小于等于阈值，则不再向下遍历并绘制该粒子[19-20]。

分形几何是近代数学发展的一个重要分支，它是研究非规则几何形态的几何学。不规则形态在自然界广泛存在，因而分形几何同时被称为是描述自然界的几何学。普遍认为 1967 年 Mandelbrot 发表的论文《英国海岸线有多长》是分形几何学正式产生的标志。然而事实上在此之前分形几何的许多重要问题早已提出，例如斯宾塞三角、康托尔集等。迭代函数系统(iterated function system, IFS)是分形几何学中一个重要且最有发展前景的分支。迭代函数系统将待生成的图形看做是由许多与整体相似的（自相似）或经过一定变换与整体相似的（自仿射）小块拼贴而成，通过对原始图形经过缩小、平移、旋转等操作绘制出新的图形，迭代函数系统生成的图形永远具有自相似性。如式(1)所示就是一个二维的仿射变换。

$$W\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \quad (1)$$

其中：W 是一个二维仿射变换，a，b，c，d，e，f 为变换的参数，通过更改这些参数，就可以得到不同的子图，以满足模拟的要求[21]。

基于点绘制的 Qsplat 绘制算法由于预处理阶段较为复杂，通常应用于大规模静态点云数据的处理，不适合动态点绘制。迭代函数系统迭代函数构造简单，便于计算机执行，对自然景观的模拟往往能取得很好的效果。通过结合两种算法，可以对传统 Qsplat 算法进行改进的到融合 IFS 的动态的 Qsplat 算法对动态点云数据进行处理。

2 三维动态积云模拟

针对积云结构紧密、孤立成块以及自相似性等特点，本文提出了基于 Qsplat 的云模拟算法。该算法使用基于 Qsplat 算法的特殊树结构粒子系统构建云体模型，然后利用 IFS 迭代函数系统丰富云体局部细节，最后利用球形图元映射二维纹理对粒子进行绘制。算法的流程如图 1。

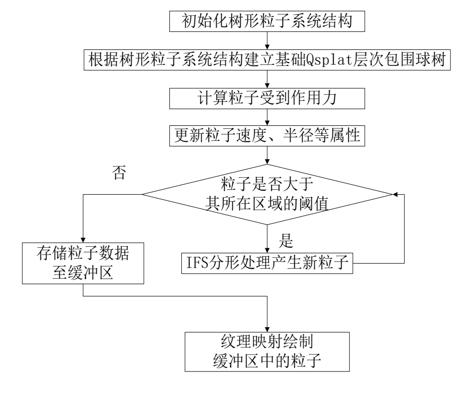


图 1 本文算法流程图

2.1 三维动态积云的建模

2.1.1 积云粒子系统的树型结构

积云的结构紧密、孤立成块且运动复杂。粒子系统方法是

最适合模拟不规则模糊动态物体的方法，但是简单的粒子系统在控制方面存在问题。针对这些问题，为了得到真实感强且易于控制的积云模型，并方便构建动态的 Qsplat 细节层次树，本文提出一种特殊的树型组织结构来构建粒子系统，并在具有父子关系的粒子间添加弹力来约束粒子的运动范围，如图 2 所示。

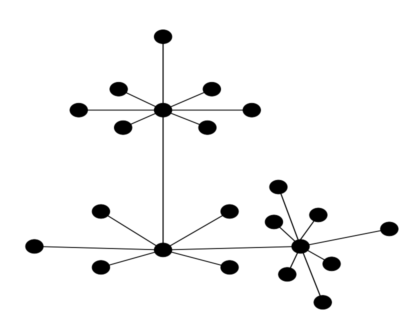


图 2　树型结构粒子系统示意图

算法首先确定根节点粒子的位置与半径(也就是云体的位置与大小)，然后在根粒子的半径内创建多个子节点粒子，每一个子节点再创建自己的七叉树子树，多个七叉树通过最终连接成一个整体。整个粒子树中根节点只有在需要移动整个云体是会产生位移，除此外不受任何力影响，其位置固定，不会随时间变化而移动。粒子系统中除根节点外的粒子都会受到重力、风力、空气阻力的影响。其中值得注意的是，除了根粒子外的粒子所受弹力仅来自于它的父节点粒子，这样可以更好地控制整个粒子系统的体积，粒子系统中的粒子（除根粒子）都有自己的半径，半径值会随时间和粒子位置的变化而发生改变。

2.1.2 积云粒子的运动方程

积云的运动主要表现为由最初的小淡积云逐渐向上、向四周生长，形成各种结构紧密的不规则的花椰菜形。基于物理方法的模拟往往具有较大的运算量，对计算机硬件具有很高的要求，不利于大规模实时模拟。因此，本文对粒子运动模型进行了简化。

根节点粒子作为锚点确定整个云体的位置与大概范围，各层次的子节点粒子收风力作用产生运动，模拟积云生长的过程，在粒子与其父节点粒子间添加弹力约束粒子运动范围、控制云体的生长不至于脱离范围导致云体空洞。本文模型中除根节点粒子外所有粒子运动过程中受重力、来自父粒子的弹力、风力以及空气阻力的作用，具体情况如图 3 所示。其中，G 是粒子所受重力，spring 表示弹力，F 表示风力，f 表示空气阻力。

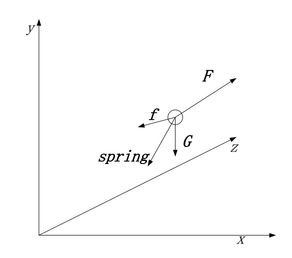


图 3　积云粒子受力分析图

为了高效真实地模拟积云的生长效果，本文提出一种简化的风场模型，模拟积云向四周扩张以及向上生长的特点。模型的风力是积云生长的动力，函数表达式如式(2)。

$$\boldsymbol{F}_{wind}=\frac{f_1\times z}{\sqrt{x^2+y^2}}\times\boldsymbol{x}+f_2\times\boldsymbol{y}+\frac{f_1\times x}{\sqrt{x^2+y^2}}\times\boldsymbol{z} \tag{2}$$

其中，f1 为风力在 x-z 平面方向分量的模，表达式如式(3)所示。

$$f_1=6.25\times10^{-4}\sqrt{x^2+z^2}+0.001\times y+0.1 \tag{3}$$

f2 为风力在 y 方向分量的模，表达式如式(4)。

$$f_2=-1.43\times10^{-4}\sqrt{x^2+z^2}-1.43\times10^{-4}y+0.004 \tag{4}$$

x、y 和 z 为三维空间坐标，通过控制常数系数的值可以改变积云在水平方向与垂直方向生长的速度。

粒子所受空气阻力为其运动的阻力之一，函数表达式如式(5)所示。

$$\boldsymbol{F}_{res}=-1.5(\boldsymbol{speedx}+\boldsymbol{speedy}+\boldsymbol{speedz}) \tag{5}$$

其中，-1.5 为空气阻力系数值，空气阻力系数值越大，粒子速度下降越快。speedx、speedy 和 speedz 分别为坐标 x、y 和 z 方向的速度分量。

粒子弹力是约束云体的主要作用力，函数表达式如式(6)所示。

$$\boldsymbol{F}_{spring}=0.05(1.0-d)^2 \tag{6}$$

其中，0.05 为弹力系数的值，改变弹力系数的值可调节积云的紧密程度，d 为当前粒子与其父粒子之间的距离，表达式如式(7)所示。

$$d=\sqrt{(x-x_0)^2+(y-y_0)^2+(z-z_0)^2} \tag{7}$$

x、y、以及 z 为当前粒子位置坐标，x0、y0 和 z0 是其父节点粒子位置坐标。

2.1.3 结合 IFS 算法的改进 Qsplat 算法

单纯的的粒子系统为达到较高的真实感往往会需要大量的粒子来提高模拟图形的精度，然而这又会降低模拟的效率。为了解决这个矛盾，本文提出了一种改进的 Qsplat 算法来保证在不降低模拟图形的精度的条件下的模拟效率。

Qsplat 算法一般被用来加快处理大规模的静态点云数据，它需要经过复杂的预处理阶段，这不利于动态实时图形绘制。因此本文对 Qsplat 算法进行改进，固化具有一定层数的基础细节层次树简化预处理，来适合本文算法的动态实时的需要。同时，算法为了提升模拟图形局部细节层次，需要为一些距离视点较近的粒子扩展细节层次，因此在基本 Qsplat 算法基础上融合 IFS 迭代函数系统，对这些粒子进行分形迭代处理，增加局部粒子，丰富模拟图形局部层次细节，提高图形的真实感。

对于给定的需要分形处理的叶子节点 M(x,y,z)，经过一次线性变换产生 i 个点 Mi(xi,yi,zi),i=0,1,⋯,7。

线性变换的表达式如式(8)。

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & j \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} m \\ n \\ l \end{bmatrix} \tag{8}$$

为使迭代收敛，每一次迭代必须是压缩变换。设根节点半径为 R，当前待分形处理粒子半径为 r，压缩因子 C 的表达式如式(9)。

$$C = R / r \tag{9}$$

联立得如式(8)的方程组，求出线性变换的相关系数，就可以得到相应的 IFS 码。

改进 Qsplat 算法具体步骤如下：

a)基于树型粒子结构建立如图 4 所示具有固定深度基础 Qsplat 细节层次树;

b)遍历 Qsplat 细节层次树。若当前粒子的半径小于阈值，停止遍历，将当前粒子放入待绘制粒子缓存区，否则继续向下遍历；若粒子深度被遮挡且深度超过阈值，停止遍历；若粒子阈值大于阈值但是当前粒子是叶子节点则转到第 c)步;

c) 基础细节层次树的细化分形处理。若当前粒子是叶子节点粒子且半径大于阈值，对当前粒子用迭代函数系统进行分形处理并生成它的临时子树，继续检查叶子节点，若满足条件则停止，否则继续执行 c)步;

d) 计算粒子的运动速度，更新粒子属性，返回 b)步。

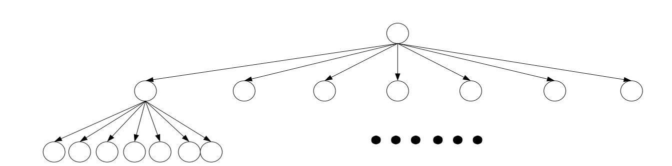


图 4　Qsplat 层次包围球结构树

2.2　三维动态积云的绘制

积云主要由浓密的球状云泡组成，大量球形气泡堆积形成各种不规则外形。考虑这个问题，本文使用的粒子以球形图元作为基础绘制单元。积云粒子为球形图元映射二维纹理图片绘制而成。同时，为了避免图元之间空洞的出现，球形图元的半径会随着粒子的运动而变化。算法使用的球形图元如图 5 所示。

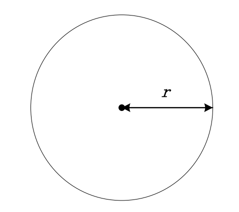


图 5　球形粒子图元示意图

粒子系统中的粒子分类以距离视点的大小为依据，处于视点周围固定距离矩形区域内的粒子利用 IFS 算法进行进一步处理，最终采用多个球形粒子映射二维纹理图片绘制；处于该区域外的粒子直接采用球形图元映射二维纹理进行绘制。

考虑积云紧密的结构及凹凸不平的外表，本文使用如图 6 所示的纹理图片来模拟积云外形。

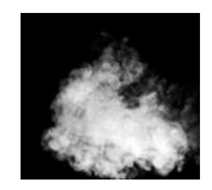


图 6　积云粒子纹理

具体的绘制步骤如下：

a) 对所有待绘制粒子根据深度由大到小排序并存储至待绘制缓存区;

b)遍历待绘制粒子缓冲区，为加快绘制速度，本文实验采用 directx9 内置球体绘制函数来预先计算球形图元数据，读得粒子数据后直接在相应位置绘制出球体;

c)利用纹理混合 Blend 等技术完成积云粒子绘制。

3　实验与分析

本文使用 VS2010 与 Directx9c 实现本文算法，所有实验在 CPU：Intel i3-2330，主频：2.4 GHz，内存：4 GB，显卡：NVS 3100M，分辨率：1280*1024 的机器上进行。

经过实验，发现模拟最低要求粒子树有两层 8 个粒子，图 7 中所示为模拟积云粒子数为 8、Qsplat 树基础层次为 2、云体初始大小 4 km 不同时刻的云模拟图形。

当粒子树为三层 57 个粒子时实验在粒子系统初始化、模拟效果以及运行帧率之间取得一个较好的平衡。图 8 中所示为模拟积云粒子数为 57 、Qsplat 树基础层次为 3、云体初始大小 4km 不同时刻的效果图。

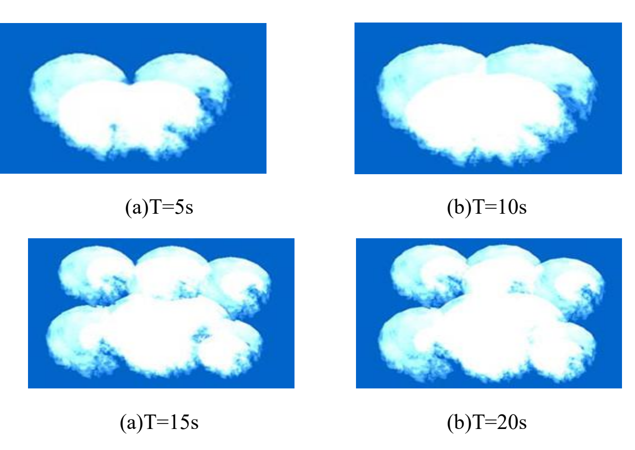
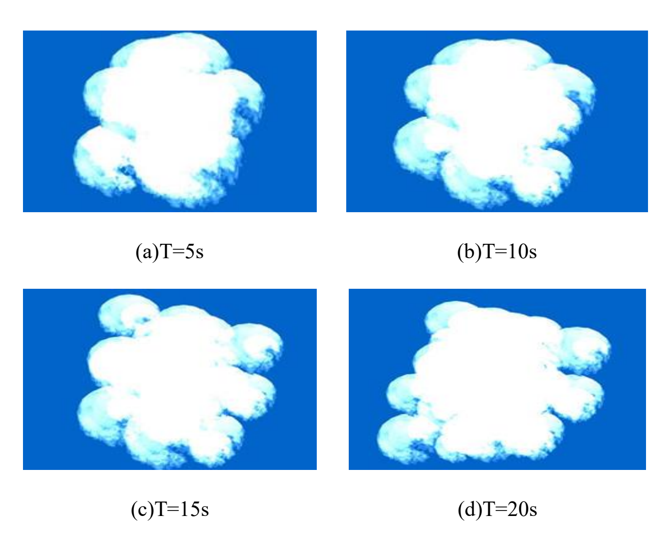


图 7　粒子数为 8 时积云模拟图形

帧率是实时算法最重要的指标之一，图 9 所示为积云粒子数为 8 和积云粒子数为 57 时模拟帧率随时间的变化曲线。由实验可知，算法运行的帧率一直保持较高的值，完全满足实时模拟的需要，并且随着时间增加，实验的帧率会逐渐稳定。

结合效果图与帧率-时间折线图分析可知，本文算法生成的云生成的云结构紧密，较好地模拟了积云的自相似性，实现了积云的向周围扩张、向上生长的运动效果，实现了三维积云的动态模拟。算法在模拟粒子树只有 2 层 8 个粒子时模拟的积云

云体粗糙，云体紧密程度较差，模拟的平均帧率为 607，在运行超过 30s 后的稳定平均帧率为 311；在模拟粒子树达到 3 层 57 个时，模拟图形云体更加紧密，云体精细度较好，模拟的平均帧率为 220，算法运行 30s 后的平均帧率为 102。由实验可知，算法可以实现较为逼真的动态积云模拟，并且可以通过增加粒子数量增加模拟云的规模和细节质量，当模拟粒子数量为 57 时，算法生成积云图形的效果较好，完全满足实时要求。



(a)T=5s　(b)T=10s

(c)T=15s　(d)T=20s

图 8　粒子数为 57 时积云模拟图形

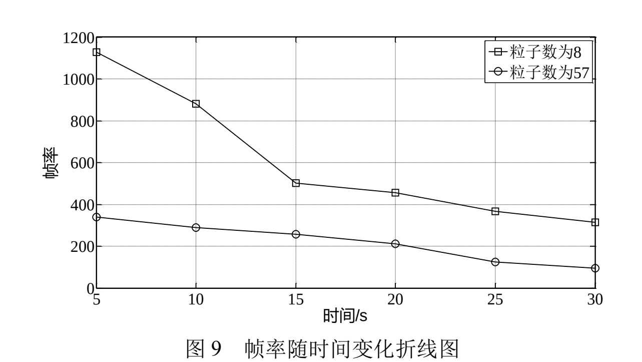


图 9　帧率随时间变化折线图

为验证改进 Qsplat 算法的加速效果，将单纯粒子系统与结合改进 Qsplat 算法的粒子系统的运行效率进行比较。实验结果的帧率随粒子数增加的折现图如图 10。单纯粒子系统的平均帧率为 126.2；结合改进 Qsplat 算法的粒子系统的平均帧率为 802.7。结果表明改进 Qsplat 算法有明显的加速效果。

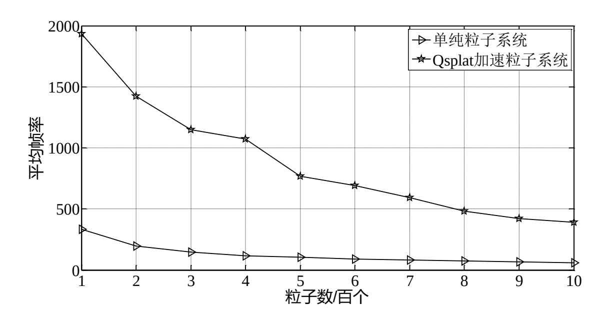


图 10　帧率随粒子数变化折线图

本文算法还与文献[5,6]提出的算法进行了比较，如表 1 所示是三种算法的基本属性情况。文献[5]的算法通过生成多个 Gradner 椭球模型结合纹理映射来实现云的模拟，但是没有实现动态效果；文献[6]通过粒子系统结合 Impostor 等技术实现了三维动态云的模拟。

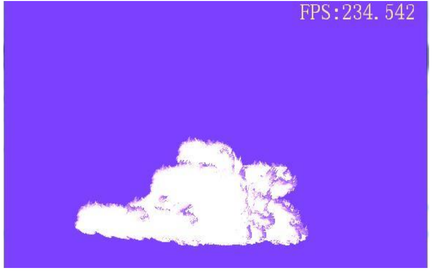
表 1 本文算法与文献[5]和文献[6]的对比

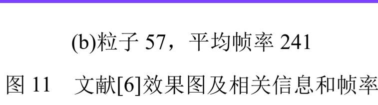
算法提出	建模方法	三维体云	是否动态	是否实时
文献[6]	Gradner 椭球模型	是	否	是
文献[8]	粒子系统	是	是	是
本文算法	树型结构 粒子系统	是	是	是

本文算法与文献[5]、文献[6]都为实时算法，为了进行算法的对比，本文对粒子数分别为 8、57 和 400 时三种算法分别进行了实验，运行效果图如图 11~13 所示（本文算法效果图均选取自帧率稳定时实验效果图）。结合图 12、13 可以看出，与文献[5]相比，粒子数分别为 8、57 和 400 时，改进 Qsplat 算法的加速比率分别为 116.55%、380.0%、1850.0%，同时改进 Qsplat 算法生成的动态云拥有更高的真实感效果。从实验结果可以看出本文算法只需要少量粒子就可以模拟云的动态效果，相比动态云模拟的文献[5]算法，本文算法的帧率有较大提升，实时性更好；结合图 11 与图 13 可以看出，与文献[6]的算法相比，虽然运行帧率不如，但是仍能完全满足实时性要求，并且相比文献[6]算法，改进 Qaplat 算法不仅实现了动态积云效果，而且生成的积云图形比文献[6]生成的云彩图形更具立体感，真实感更强。



(a)粒子数 8，平均帧率 802



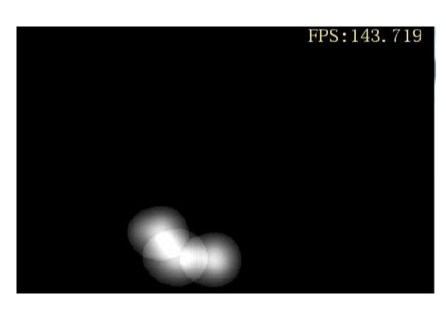


(b)粒子 57，平均帧率 241

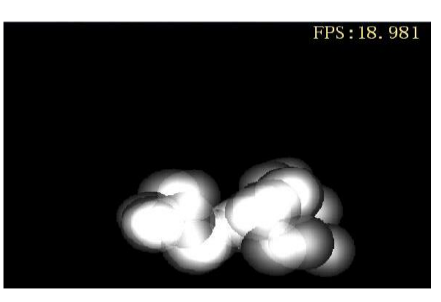


(c) 粒子 400，平均帧率 78

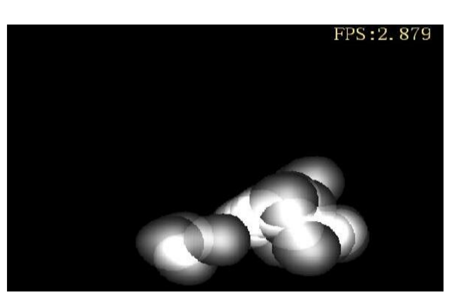
图 11　文献[6]效果图及相关信息和帧率



(a) 粒子数 8，平均帧率 145



(b) 粒子数 57，平均帧率 20



(c) 粒子数 400，平均帧率 4

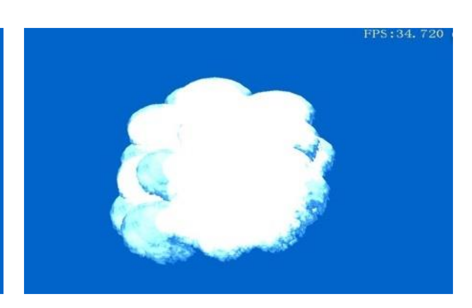
图 12　文献[8]效果图及相关信息和帧率



(a) 粒子数 8,帧率 314



(b)粒子数 57，帧率 96



(c) 粒子数 400，帧率 40

图 13　本文算法效果图及相关信息和帧率

4　结束语

积云的不定型外形给模拟带来了很大的阻碍，如何在实时性与模拟的真实感之间找到平衡是积云实时模拟的难点。本文提出了一种基于动态 Qsplat 算法的树型粒子系统积云模拟算法，在保证模拟精度的条件下尽可能地提高模拟效率，算法提出的特殊树型粒子结构粒子系统能够快速建立可控的积云结构模型，并建立动态层次包围球树加快绘制速度，丰富积云局部细节。实验结果表明，算法在保证模拟图形真实感的基础上，实现了三维动态积云的快速实时模拟，可应用于飞行场景仿真、游戏制作等领域。

本文没有对云的光照问题进行深入研究，因此无法实现真实光照条件下云彩色彩的变化，改进 Qsplat 算法的结构特点限制了该算法的运用场景，只能用于云等运动不剧烈的物体模拟。这些问题都是后续研究的重点。

参考文献：

[1] 王帅辉, 韩志刚, 姚志刚, 等. 基于 CloudSat 资料的中国及周边地区各类云的宏观特征分析 [J]. 气象学报, 2011, 69 (5): 883-899.

[2] Harris M J, Baxter W V, Scheuermannn T, et al. Simulation of dynamics on graphics hardware [C]// Proc of ACM SIGGRAPH//EUROGRAPHICS Conference on Graphics Hardware. 2003: 92-101.

[3] 唐勇, 赵丽文, 吕梦雅. 三维动态云模拟 [J]. 小型微型计算机系统, 2013, 34 (3): 659-662.

[4] 何晓曦, 陈雷霆, 朱清新. 一种简化的流体方法快速仿真大型三维云场景 [J]. 计算机应用研究, 2012, 29 (6): 2357-2359.

[5] 贺怀清, 刘浩瀚, 刘金星, 等. 一种改进的立体云模拟方法 [J]. 系统仿真学报, 2008, 20 (10): 2620-2624.

[6] 潘秋羽, 毕硕本, 陆良虎, 等. 基于粒子系统三维动态云的快速仿真算法 [J]. 系统仿真学报, 2014, 26 (1): 85-90.

[7] Elhaddad A, Elhaddad F, Sheng B, et al. Real-time cloud simulation using lennard-jones approximation [C]// Proc of International Conference on Computer Animation and Social Agents. New York: ACM Press, 2016: 131-137.

[8] 范晓磊, 张立民, 柳颖. 基于动态网格的云的实时仿真 [J]. 计算机应用研究, 2015, 32 (7): 2234-2237.

[9] Yusov E. High-performance rendering of realistic cumulus clouds using pre-computed lighting [C]// High Performance Graphics. Eurographics Association, 2014: 127-136.

[10] Dobashi Y. Inverse Approach for Visual Simulation of Clouds [M]// Mathematical Progress in Expressive Image Synthesis I. [S. l.] : Springer, 2014: 85-91.

[11] Dobashi Y, Iwasaki K, Yue Y, et al. Visual simulation of clouds [J]. Visual Informatics, 2017, 1 (1): 1–8.

[12] Goswami, Prashant, Neyret, et al. Real-time landscape-size convective clouds simulation [C]// Proc of the 19th Symposium on Interactive 3D Graphics and Games. 2015: 135-135.

[13] Rui P D, Gomes A, Rui P D, et al. Real-Time simulation of cumulus clouds through SkewT//LogP diagrams [J]. Computers & Graphics. 2017, 67: 103-114.

[14] Du Peng, Zhao Jieyi, Pan Wanbin, et al. GPU accelerated real-time collision handlingin virtual disassembly [J]. Journal of Computer Science and

Technology, 2015, 30 (3): 511-518.

[15] 王春华，李留青. 基于GPU的风中树木物理动画 [J]. 计算机应用研究, 2010, 27 (10): 3988-3990.

[16] 吕梦雅，陶建新，唐勇，等. 扩展粒子系统实时绘制逼真的龙卷风 [J]. 小型微型计算机系统, 2017, 38 (5): 1107-1110.

[17] 王迎瑞，黎雷生，王景焘，等. 光滑粒子流体动力学方法的高效异构加速 [J]. 计算机学报, 2017 (9): 2040-2056.

[18] Barbosa C W F, Dobashi Y, Yamamoto T. Adaptive cloud simulation using position based fluids [J]. Computer Animation & Virtual Worlds, 2015, 26 (3-4): 367–375.

[19] Rusinkiewicz S, Levoy M. QSplat_: a multiresolution point rendering system for large meshes [C]// Proc of SIGGRAPH. 2000: 343-352.

[20] Cura R, Perret J, Paparoditis N. Implicit LOD for processing, visualisation and classification in Point Cloud Servers [DB//OL]. arXiv. 2016.

[21] Quirce J, Gálvez A, Iglesias A. Computing self-similar contractive functions for the ifs inverse problem through the cuckoo search algorithm [C]// Proc of International Conference on Harmony Search Algorithm. 2017: 333-342.